

REMARKS

Claims:

Claims 1, 2, 7-11, 16-18, 23, 24, 28, 29 and 35 comprise the case, Claims 3-6, 12-15, 19-22, 25-27 and 30-34 having been canceled hereby.

The claims presently comprising the case have been amended in accordance with the specification to incorporate the language "boot program code" as relating to the "minimum operational state", that the "minimally operational state" comprises "only boot program code", and to incorporate the language "operating" for the "code image required to become fully operational". Applicant respectfully submits that the language "boot program code" is used throughout the specification, that "operating" is intimated throughout the specification, that the term "only contains the boot code" is at page 18, line 23, and is similar to the language at page 9, lines 17-20. Hence, Applicant respectfully submits that no new matter has been added.

35 U.S.C 103(a)

I) Claims 1, 2, 9-11, 18, 23, 24, 29 and 35:

Of the claims remaining in the case, Claims 1, 2, 9-11, 18, 23, 24, 29 and 35 stand rejected under 35 U.S.C 103(a) as being unpatentable over US Patent No. 6,023,727 to Barrett et al. in view of US Patent No. 6,269,396 to Shah et al. Claims 16, 17 and 28 were also mentioned, but the actual rejections thereof follow.

**A) Claims 1, 10, 23 and 35:**

The Examiner rejected Claims 1 and 35 together, rejected Claim 10 under the same reason, as the "method corresponding to Claim 1", and rejected Claim 23 on essentially the same grounds, adding a "non-volatile memory" of Barrett. The Examiner presents Barrett and Smith separately:

**1 Barrett:**

**a)** With respect to Barrett, the Examiner states that Barrett discloses "a network \*\*\*; a plurality of processors coupled in said network \*\*\* having a minimally operational state \*\*\*; and having a fully operational state employing a code image".

**Barrett teaches away from Applicant's Claims 1, 10, 23 and 35:**

The Examiner fails to cite any language in Barrett indicating a minimally operational state different than a fully operational state.

As pointed out by the previously submitted Declaration under Rule 1.132, Barrett describes a "reprogrammable network communication device" which does not have a minimally operational state, and so has only a fully operational state, described as 'NEB 101 is shipped with operational software' \*\*\*. See also, column 7, lines 56-61, where the software is executed by a microprocessor out of 'flash EPROM 174' or can be 'selectively moved to the higher performance 512 KB DRAM 175'.'" (Emphasis added). Thus, Barrett reprograms an existing fully operational state.

To better emphasize the difference between the minimally operational state and the fully operational state, Applicant has amended the claims to recite, e.g. Claim 1, "a network; a plurality of processors coupled in said network, each of said

processors comprising a non-volatile memory configured to store program code of a minimally operational state, said program code comprising only boot program code, said minimally operational state absent an operating code image required to become fully operational, said boot program code sufficient to operate said processor to provide a code image request; and comprising a volatile memory configured to store said operating code image, said operating code image configured to place said processor in a fully operational state; \*\*\*". (Emphasis added).

Hence, Applicant respectfully submits that Barrett teaches away from Applicant's claims.

**b)** Further with respect to Barrett, the Examiner states that Barrett discloses "said processors \*\*\* requesting said code image from said network \*\*\* 'network administrator's PC 103, the network administrator can remotely alter the ROM firmware image in flash EPROM 174 by downloading new data' and \*\*\* 'The program respond to requests... for data download')".

**Barrett teaches away from Applicant's Claims 1, 10, 23 and 35:**

In the statement above, it is clear that the network administrator downloads new data to the flash EPROM 174 of the network interface cards, the network interface cards do not request the data.

As further pointed out by the previously submitted Declaration under Rule 1.132, "That, the Barrett reprogrammable network communication device does not request the code image from the network. Instead, the code image already exists in 'flash EPROM 174' \*\*\*.

"That, the code image supplied by Barrett is conducted by the network administrator at (his) own initiative, and not at the request of a processor in the network. Specifically, 'from the network administrator's PC 103, the network administrator can

remotely alter the ROM firmware image in flash EPROM 174 by downloading new data' \*\*\*.

"The Barrett administrator insures that compatible images are downloaded by scanning the network to identify targets, and has 'software code which ensures that the downloaded image is compatible before actual reprogramming occurs' \*\*\*."

Hence, Applicant submits that Barrett teaches away from Applicant's claims, e.g. Claim 1, "\*\*\*\* said processors, when in said minimally operational state, employing said boot program code to request said operating code image from said network by means of said code image request; \*\*\*\*". (Emphasis added).

c) Still further with respect to Barrett, the Examiner states that Barrett discloses "a master source \*\*\*, upon receiving said code image request waiting a predetermined time period \*\*\* allowing any additional said processor to reach said minimally operational state \*\*\* 'NEB microprocessor 173 stops writing to memory \*\*\* at predetermined intervals allowing printer interface microprocessor 151 sole access to the memory until it catches up', broadcasting said code image on said network (\*\*\* 'proper image is sent to the targeted NEB')."

**Barrett teaches away from Applicant's Claims 1, 10, 23 and 35:**

In the statement above, it is clear that the NEB resends information to its associated printer, and that any wait is to allow the printer interface to catch up. Applicant submits that this is unrelated to Applicant's predetermined time period to allow any additional said processor to reach the minimally operational state.

As pointed out by the previously submitted Declaration under Rule 1.132, "The Barrett administrator insures that compatible images are downloaded by scanning the network to identify

targets, and has 'software code which ensures that the downloaded image is compatible before actual reprogramming occurs' \*\*\*.

"Barrett does not wait a predetermined time period, the predetermined time period allowing any additional processor to reach the minimally operational state, and does not, upon completion of the predetermined time period, show a master source broadcasting the code image on the network, as is done by the present '917 Application."

Additionally, Applicant points out that the Examiner has referred to the microprocessor 173 of NEB 101 as waiting at predetermined intervals for a catch up, as though the NEB is a master processor waiting for a predetermined time period as per Applicant's claims, but the NEB is, instead, a target, and is controlling the timing for writing to its printer's memory.

Hence, Applicant submits that Barrett teaches away from Applicant's claims, e.g. Claim 1, "\*\*\*\* a master source coupled in said network, said master source configured to provide at least said operating code image for broadcasting said operating code image on said network, said master source, upon receiving said code image request, waiting a predetermined time period, said predetermined time period allowing any additional said processor to reach said minimally operational state, and, upon completion of said predetermined time period, broadcasting said operating code image on said network." (Emphasis added).

d) With respect to Barrett, the Examiner states that Barrett "does not explicitly disclose said minimally operational state absent a code image to required to become e fully operational \*\*\*."

**Barrett teaches away from Applicant's Claims 1, 10, 23 and 35:**

Applicant agrees with the Examiner that Applicant's claims differentiate from Barrett.

**2     Shah:**

e) With respect to Shah, the Examiner states that Shah discloses a "minimally operational state (OS\_MIN\*\*\* sufficient to provide a code image request (\*\* 'Brings up server... fully operational state... form a minimal operational state... Enable/Disable... processing... server node').".

**Shah teaches away from Applicant's Claims 1, 10, 23 and 35:**

Applicant submits that the terminology of Shah does not relate to a truly minimally operational state as defined by Applicant's claims, which comprises, e.g. Claim 1, "program code of a minimally operational state, said program code comprising only boot program code, said minimally operational state absent an operating code image required to become fully operational" (emphasis added). Rather "OS\_MIN" is defined by Shah at column 4, lines 39-48 as a configurable attribute of a configurable element, the configurable attributes including "RunLevel, which is the level a configurable element starts at. The RunLevels include PRE\_MIN, OS\_MIN, \*\*\*." At column 5, lines 31-47, Shah states "A node 24 is defined as an instance of a supported operating system on which telecom platform 10 runs. Telecom platform 10 provides software that manages processes on nodes 24. \*\*\* Nodes 24 have operating states, supported by telecom platform, that describe the ordering of configurable elements started within them. \*\*\* The OS\_MIN node state coordinates all configurable elements configured for the OS\_MIN run level will be started to bring the node to the OS\_MIN state."

Hence, the Shah OS\_MIN is a run level where the operating system is already in place, and is clearly not program code of a minimally operational state, said program code comprising only boot program code, where the minimally operational state is absent an operating code image required to become fully operational.

Further, a node running at OS\_MIN cannot request an operating code image required to become fully operational, since the Shah node is already the "supported operating system on which telecom platform 10 runs".

Still further, per column 6, lines 35-36, Shah states "Each node has a single instance of the operating system running on it." Thus there is no need to have Applicant's claimed minimally operational state, said program code comprising only boot program code.

Hence, Applicant submits that Shah teaches away from Applicant's claims, e.g. Claim 1, "\*\*\* program code of a minimally operational state, said program code comprising only boot program code, said minimally operational state absent an operating code image required to become fully operational, said boot program code sufficient to operate said processor to provide a code image request; and comprising a volatile memory configured to store said operating code image, said operating code image configured to place said processor in a fully operational state \*\*\*." (Emphasis added).

### 3 Barrett in view of Shah:

f) With respect to Barrett and Shah, the Examiner states that "it would have been obvious \*\*\* to incorporate the method of bringing up server minimal operational state or fully operational

state to enable/disable the processing as taught by Shah into the method of distributing the code image as taught by Barrett."

**Barrett in view of Shah teaches away from Applicant's Claims 1, 10, 23 and 35:**

Applicant submits that, as pointed out above, the Shah OS\_MIN is a run level, where the operating system is already in place, and is not Applicant's e.g. Claim 1, "\*\*\* program code of a minimally operational state, said program code comprising only boot program code, said minimally operational state absent an operating code image required to become fully operational, \*\*\*" (emphasis added), and that Shah does not request code; and that Barrett relates to a distribution system operated by a network administrator who downloads new data to the flash EPROM of the network interface cards, the network interface cards do not request the data, essentially opposite to Applicant's claimed invention, and that Barrett teaches away from Applicant's claimed invention for the additional stated reasons.

Therefore, taken together, Applicant respectfully submits that Barrett and Shah clearly teach away from Applicant's claimed invention.

Applicant thus respectfully requests allowance of Claims 1, 10, 23 and 35 under 35 U.S.C. 103(a).

**B) Claims 2, 11 and 24:**

The Examiner rejected Claim 2, rejected Claim 11 under the same reason as the "method corresponding to Claim 2", and rejected Claim 24 on essentially the same grounds.



g) The Examiner states that Barrett discloses "receive and implement said code image" but "Barrett does not explicitly disclose said minimally operational state. However, Shah discloses \*\*\* 'brings up the server node to minimally operational state (OS\_MIN')" which "would be obvious for the reasons set forth in the rejection of claim 1."

Barrett in view of Shah teaches away from Applicant's Claims 2, 11 and 24:

Applicant submits that, as pointed out above, the Shah OS\_MIN is a run level, where the operating system is already in place, and is not program code of a minimally operational state, said program code comprising only boot program code, where the minimally operational state is absent an operating code image required to become fully operational. Similarly, as discussed above, Barrett describes a "reprogrammable network communication device" which does not have a minimally operational state, and so has only a fully operational state. Thus, Barrett reprograms an existing fully operational state.

Therefore, taken together, Applicant respectfully submits that Barrett and Shah clearly teach away from Applicant's claimed invention, e.g. Claim 2, "wherein said processors, additionally, upon said broadcast of said operating code image, employ said boot program code to receive and implement said operating code image only if said processor is in said minimally operational state." (Emphasis added).

Further, Claims 2, 11 and 24 depend respectively from Claims 1, 10, and 23 and are therefore also submitted to be patentable over Barrett in view of Shah.

Applicant thus respectfully requests allowance of Claims 2, 11 and 24 under 35 U.S.C. 103(a).

**C) Claims 9, 18 and 29:**

The Examiner rejected Claim 9, rejected Claim 18 under the same reason as the "method corresponding to Claim 9", and rejected Claim 29 on essentially the same grounds.

h) The Examiner states that Barrett discloses "\*\*\*\* select said broadcast code image for implementation if said determination determines that said code image is correct for said processor (\*\* 'microprocessor... downloads the new image into DRAM \*\*\* reprogram... ERPOM... only... compatibility is confirmed')."

**Barrett in view of Shah teaches away from Applicant's Claims 9, 18 and 29:**

Applicant submits that, as pointed out above, the Shah OS\_MIN is a run level, where the operating system is already in place, and is not program code of a minimally operational state, said program code comprising only boot program code, where the minimally operational state is absent an operating code image required to become fully operational. Similarly, as discussed above, Barrett describes a "reprogrammable network communication device" which does not have a minimally operational state, and so has only a fully operational state. Thus, Barrett reprograms an existing fully operational state.

Therefore, taken together, Applicant respectfully submits that Barrett and Shah clearly teach away from Applicant's claimed invention, e.g. Claim 9, "wherein said master source is configured to provide a plurality of different said operating code images, wherein said processor requesting said operating code image requests one of said different operating code images, wherein said master source is configured to broadcast said requested one of said different operating code images, and

wherein said processors employ said boot program code additionally to determine whether said broadcast operating code image is correct for said processor, and to select and store in said volatile memory, said broadcast operating code image for implementation if said determination determines that said operating code image is correct for said processor." (Emphasis added).

Further, Claims 9, 18 and 29 depend respectively from Claims 1, 10, and 23 and are therefore also submitted to be patentable over Barrett in view of Shah.

Applicant thus respectfully requests allowance of Claims 9, 18 and 29 under 35 U.S.C. 103(a).

**II) Claims 7, 8, 16, 17 and 28:**

Of the claims remaining in the case, Claims 7, 8 and 28 stand rejected under 35 U.S.C 103(a) as being unpatentable over Barrett in view of Shah and further in view of Harmer et al., US Patent No. 6,401,198. Claims 16 and 17 are rejected under the same reason, as the method corresponding to Claims 7 and 8.

i) The Examiner states that the rejection of Claims 1, 10 and 23 are incorporated, but that "neither Barrett nor Shah explicitly discloses code image is a combination of different images."

"However Harmer discloses one code image contains multiple code images (\*\* 'one code image making up the first portion...and second portion... of... BIOS' and \*\* 'BIOS... include multiple images... each... images corresponding to a different...computer architecture... device... attached')."

"Therefore, it would have been obvious \*\*\* to incorporate the method of having one code image with different images included as taught by Harmer into the method of distributing the code image as taught by the combination system of Barrett and Shah."

**Barrett in view of Shah and in view a Harmer teaches away from Applicant's Claims 7, 8, 16, 17 and 28:**

Applicant submits that, as pointed out above, the network administrator of Barrett downloads new data to the flash EPROM 174 of the network interface cards, the network interface cards do not request the data.

As further pointed out by the previously submitted Declaration under Rule 1.132, "That, the code image supplied by Barrett is conducted by the network administrator at (his) own initiative, and not at the request of a processor in the network. Specifically, 'from the network administrator's PC 103, the network administrator can remotely alter the ROM firmware image in flash EPROM 174 by downloading new data' \*\*\*."

Applicant submits that, as discussed above, Shah does not relate to a truly minimally operational state as defined by Applicant's claims. Rather, the Shah OS\_MIN is a run level where the operating system is already in place, and is clearly not program code of a minimally operational state, said program code comprising only boot program code, where the minimally operational state is absent an operating code image required to become fully operational.

Hence, a node running at OS\_MIN cannot request an operating code image required to become fully operational, since the Shah node is already the "supported operating system on which telecom platform 10 runs".

Still further, per column 6, lines 35-36, Shah states "Each node has a single instance of the operating system running on it." Thus there is no need to have Applicant's claimed minimally operational state, said program code comprising only boot program code.

Harmer also has no minimally operational state, and instead has only a fully operational state stored within the "computer system", which comprises a BIOS, with "at least a portion of the BIOS to be stored within the mass memory storage of a mass memory storage peripheral computer device rather than being stored within ROM." (Abstract, lines 2-10). Harmer thus also does not have a minimally operational state and does not request the fully operational state code image from the network.

Hence, Applicant respectfully submits that the combination of Barrett, Shah and Harmer teach away from Applicant's invention, e.g. Claim 7, "wherein said master source provides one said operating code image for any said code image request" (emphasis added); and from, e.g. Claim 8, "The multi-node network of processors of Claim 7, wherein ones of said processors implement different said operating code images, wherein said one master source operating code image comprises a combination of said different operating code images, and wherein said processors employ said boot program code additionally to select, store in said volatile memory, and implement one of said combination of different operating code images." (Emphasis added).


Applicant thus respectfully requests allowance of Claims 7, 8, 16, 17 and 28 under 35 U.S.C. 103(a).

Additional Art:

The art cited by the Examiner in addition to that previously cited and that cited above have been examined and as best understood, do not teach or suggest Applicant's claimed invention. The Examiner cited USPN 5,483,656, Oprescu et al.; USPN 5,752,046, Oprescu et al.; and USPN 5,842,027, Oprescu et al. Applicant submits that none of the cited patents teach, either singly or in combination, the present invention as described and claimed in Applicant's Claims 1, 2, 7-11, 16-18, 23, 24, 28, 29 and 35.

Accordingly, Applicant believes the present invention distinguishes over the cited patents and respectfully requests that the Examiner allow Applicant's Claims 1, 2, 7-11, 16-18, 23, 24, 28, 29 and 35 under 35 U.S.C. 103.

Respectfully submitted,  
B. G. Goodman

By:   
John H. Holcombe, (#20,620)  
Attorney for Applicants  
From: IBM Corporation  
Intellectual Property Law  
8987 E. Tanque Verde Rd. #309-374  
Tucson, AZ 85749  
Telephone: (520) 760-6629

JHH/cw